**DOCUMENTAZIONE PROGETTO PARCHEGGIO**

1. **Descrizione teorica**

Il parcheggio è un automa con due stati creato con Arduino. Un automa è una macchina formata da un numero finito di stati. Questo automa ha 11 stati: da 0 (che consiste nel parcheggio vuoto) a 10 (cioè parcheggio pieno). Per spostarsi da uno stato all’altro si sfruttano l’ingresso e l’uscita delle auto. Se un’auto esce ci si avvicina allo stato 0, se un’auto entra ci si avvicina allo stato 10.

1. **Arduino**

Arduino è un microcontrollore programmabile grazie all’omonimo software. Questo ci permette di controllare tutte le componenti usate per la creazione del parcheggio (come servo motori e luci), tramite dei pin ai quali si può far arrivare un segnale tramite il software.

1. **Componentistica utilizzata**

Il parcheggio per funzionare utilizza due servo motori, due fotoresistenze e tre led, il tutto ovviamente collegato tramite dei fili ad arduino che ne permette l’alimentazione ed il funzionamento programmato.

1. **Software commentato**

Per la realizzazione del parcheggio abbiamo usato un microcontrollore di nome Arduino. Per questo il linguaggio di programmazione è appunto quello di Arduino.  
Nel progetto sono presenti 2 servomotori e per questo abbiamo importato la libreria Servo.h ed è presente anche un Display LCD, per il quale abbiamo usato la libreria LiquidCrystal.h.  
Inoltre abbiamo aggiunto tutte le variabili necessarie per il funzionamento, ad esempio variabili per l’input e contatori.

#include <Servo.h> Libreria per far funzionare i servo motori

#include <LiquidCrystal.h> Libreria per far funzionare il display LCD

LiquidCrystal lcd(12,11,9,8,7,6); Si indicano i pin a cui abbiamo collegato il display

Servo servo; Dichiariamo la variabile servo di tipo Servo per i primo servo motore

Servo servo2; Dichiariamo la variabile servo2 di tipo Servo per i secondo servo motore

int rosso=4; rosso è la variabile che indica il led rosso del semaforo, collegato al pin 4

int verde=2; verde è la variabile che indica il led verde del semaforo, collegato al pin 2

int giallo=3; giallo è la variabile che indica il led giallo del semaforo, collegato al pin 3

int i;

int analogInPin3 = A4;

int analogInPin = A0; variabile legata al pin A0, dove abbiamo collegato il sensore per l’entrata dell’auto

int analogInPin2 = A1; variabile legata al pin A1, dove abbiamo collegato il sensore per l’uscita dell’auto

int sera=0;

int distanzaentrata = 0; variabile che ci indica se un’auto entra

int distanzauscita = 0; variabile che ci indica se un’auto esce

int cont;

int lib; contatore dei posti liberi

int occ; contatore dei posti occupati

Nel void setup abbiamo indicato il pin nei quali era collegato ciascun sensore e dispositivo e abbiamo inizializzato le variabili.

void setup() {

pinMode(rosso,OUTPUT); pinMode è un’istruzione che ci permette di indicare se un pin è di

pinMode(analogInPin,INPUT); OUTPUT (Arduino manda il segnale) o di tipo INPUT (Arduino riceve

pinMode(analogInPin2,INPUT); il segnale)

pinMode(analogInPin3,INPUT);

pinMode(10,INPUT);

pinMode(rosso,OUTPUT);

pinMode(giallo,OUTPUT);

pinMode(verde,OUTPUT);

pinMode(13,OUTPUT);

i=1;

cont=0;

occ=0;

lib=10;

servo.attach(13); Indichiamo a quale pin abbiamo collegato il primo servo

servo2.attach(19); Indichiamo a quale pin abbiamo collegato il secondo servo

Serial.begin(9600); Mettiamo in funzione il monitor seriale

lcd.begin(16,2); Gestione del display LCD

lcd.print("Parcheggio ");

lcd.setCursor(0,1);

lcd.print("GALILEO GALILEI");

delay(2000);

lcd.clear();

}

Nel void loop invece abbiamo creato il funzionamento del programma.

void loop() {

digitalWrite(rosso,HIGH); digitalWrite è un’istruzione per aumentare o diminuire il voltaggio in un pin

lcd.clear();

lcd.setCursor(6,1);

lcd.print("PIENO ");

servo.write(10);

while(occ <=9){

servo.write(-90); Settaggio posizione iniziale del primo servo motore

servo2.write(147); Settaggio posizione iniziale del secondo servo motore

distanzaentrata = analogRead(analogInPin);

distanzauscita = analogRead(analogInPin2);

sera = analogRead(analogInPin3);

lcd.clear();

digitalWrite(rosso,HIGH);

lcd.setCursor(0,0);

lcd.print("Post liberi= ");

lcd.setCursor(13,0);

lcd.print(lib);

lcd.setCursor(15,0);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Post occupati= ");

lcd.setCursor(14,1);

lcd.print(occ);

Serial.println(distanzaentrata);

if(sera>=70){

digitalWrite(13,HIGH);

}

else{

digitalWrite(13,LOW);

}

if(distanzaentrata <10){ Entrata di un’auto

lib=lib-1; Decremento i posti liberi

occ=occ+1; Incremento i posti occupati

digitalWrite(rosso,LOW);

digitalWrite(giallo,LOW);

cont=31;

for(i=1;i<=13;i++){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("ATTENDERE...");

cont=cont+5;

servo.write(cont); Movimento del servo motore per alzare la sbarra

digitalWrite(giallo,LOW);

delay(250);

digitalWrite(giallo,HIGH);

delay(250);

if(cont>=96){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("PROCEDA...");

distanzaentrata = analogRead(analogInPin);

distanzaentrata=0;

digitalWrite(giallo,LOW);

digitalWrite(verde,HIGH);

servo.write(120); Movimento del servo motore per abbassare la sbarra

delay(5000);

digitalWrite(verde,LOW);

}

}

}

if(distanzauscita <10){ Uscita di un’auto

lib=lib+1; Incremento i posti liberi

occ=occ-1; Decremento i posti occupati

cont=157;

digitalWrite(rosso,LOW);

digitalWrite(giallo,LOW);

for(i=1;i<=5;i++){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("ATTENDERE...");

cont=cont-5;

servo2.write(cont); Movimento del servo motore per alzare la sbarra

digitalWrite(giallo,LOW);

delay(250);

digitalWrite(giallo,HIGH);

delay(250);

if(cont==132){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("PROCEDA...");

digitalWrite(giallo,LOW);

digitalWrite(verde,HIGH);

servo2.write(50); Movimento del servo motore per abbassare la sbarra

delay(5000);

digitalWrite(verde,LOW);

}

}

}

else{

servo2.write(5);

}

}

}

Classe 3D

Prof. Marone

Prof. Naddeo